

Assignment #B: Dec Mock Exam大雪前一天

Updated 1649 GMT+8 Dec 5, 2024

2024 fall, Compiled by 昂奕,化学与分子工程学院

说明:

- 1) 月考: AC2(啊啊啊受不了了)。考试题目都在“题库(包括计概、数算题目)”里面,按照数字题号能找到,可以重新提交。作业中提交自己最满意版本的代码和截图。
- 2) 请把每个题目解题思路(可选),源码Python,或者C++(已经在Codeforces/Openjudge上AC),截图(包含Accepted),填写到下面作业模版中(推荐使用typora <https://typoraio.cn>,或者用word)。AC或者没有AC,都请标上每个题目大致花费时间。
- 3) 提交时候先提交pdf文件,再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、“作业评论”区有上传的md或者doc附件。
- 4) 如果不能在截止前提交作业,请写明原因。

1. 题目

E22548: 机智的股民老张

<http://cs101.openjudge.cn/practice/22548/>

思路: 我真以为这是个E题...

从最后一天的股价往前看,

如果某天的股价大于后面几天最大的股价就更新"后面几天股价"的最大值,

反之用后面几天股价的最大值减去它来更新最大利润

13min

代码:

```
v=list(map(int,input().split()))
v.reverse()
mi=99999
ma=0
p=0
for i in range(len(v)-1):

    if v[i+1]>ma:
        ma=v[i+1]
    if ma-v[i+1]>p:
        p=ma-v[i+1]
print(p)
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
v=list(map(int,input().split()))
v.reverse()
dp=[0]*100002
mi=99999
ma=0
p=0
for i in range(len(v)-1):

    if v[i+1]>ma:
        ma=v[i+1]
    if ma-v[i+1]>p:
        p=ma-v[i+1]
print(p)
```

基本信息

#: 47566661
题目: E22548
提交人: 24n2400011782
内存: 9592kB
时间: 40ms
语言: Python3
提交时间: 2024-12-05 15:21:22

M28701: 炸鸡排

greedy, <http://cs101.openjudge.cn/practice/28701/>

思路：最理想情况是每块鸡排都炸完,用时为所有鸡排用时的平均值
如果耗时最长的鸡排比其他鸡排的平均用时更长,说明无论其他鸡排怎么摆都会使那块鸡排炸不完,
那索性那块鸡排就一直放那炸,考虑剩下的鸡排
否则就可以达到最理想情况:平均用时
代码：

```
n,k=map(int,input().split())
jp=list(map(int,input().split()))
s=sum(jp)
jp.sort(reverse=True)
def f(ma,s,k):

    if k>1 and jp[ma]>(s-jp[ma])/(k-1) :
        return (f(ma+1,s-jp[ma],k-1))
    elif k>0:
        return (s/k)

print('{:.3f}'.format(f(0,s,k)))
```

代码运行截图 == （至少包含有"Accepted"） ==

状态: Accepted

源代码

```
n,k=map(int,input().split())
jp=list(map(int,input().split()))
s=sum(jp)
jp.sort(reverse=True)
def f(ma,s,k):

    if k>1 and jp[ma]>(s-jp[ma])/(k-1) :
        return (f(ma+1,s-jp[ma],k-1))
    elif k>0:
        return (s/k)

print('{:.3f}'.format(f(0,s,k)))
```

基本信息

#: 47568104
题目: M28701
提交人: 24n2400011782
内存: 3848kB
时间: 24ms
语言: Python3
提交时间: 2024-12-05 16:12:47

M20744: 土豪购物

dp, <http://cs101.openjudge.cn/practice/20744/>

思路：第一遍硬dp爆内存，第二遍用了滚动数组超时($O(n^2)$)

然后看题解

没想到还有这样高的方法

记 $dp[i]$ 为前 i 个商品，不放回，可获得最大价值。

$v[i]$ 为商品 i 的价值

商品必然可分为如下几段：

一段内所有商品价值和 ≥ 0 (记为+)

一段内所有商品价值和 < 0 (记为-)



当考虑相邻两段合并时，又可以合成一段+或-



如果不考虑放回，应有如下策略：

1> $[-] [+]$ < $[-+]$ 如果前一段为负，不如不拿这段，直接考虑后面。

2> $[+] [-] [+]$ 拿 $+ - +$ 交替应先将 $+ -$ 合并再考虑。



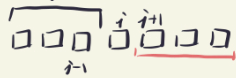
于是有 $dp[i] = \max(dp[i-1] + v[i], v[i])$

如果 $dp[i-1]$ 原来为负，那就对应 1>

如果 $dp[i-1]$ 原来为非负，那不管 $v[i]$ 的正负，加上再说

下面考虑放回.

"放回"第*i*个不妨视作以第*i*个为终止和以第*i*个为起始同时取.



即"放回"第*i*个的最大价值 = 以第*i*个为终止的最大价值 + 以第*i*个为起始的最大价值
故类似于 dp1 构建 dp2.

这种计算针对的是选 *n* 放 1' (*n* ≥ 3) 的,
但实际"选 2 放 1" 已含在 dp1 的结果内, 不必再考虑.

对于不放回, 取 $ans = \max(dp1)$

对于放回, 从 1 ~ *n* - 2 的每个 *i*, $ans = \max(ans, dp1[i] + dp2[i+1])$.

代码:

```
v=list(map(int,input().split(',')))
n=len(v)
dp1=[-float('inf')]*n
dp2=dp1[:]
dp1[0]=v[0]
dp2[-1]=v[-1]
for i in range(1,n):
    dp1[i]=max(dp1[i-1]+v[i],v[i])
    dp2[n-i-1]=max(dp2[n-i]+v[n-i-1],v[n-i-1])
ans=max(dp1)
for i in range(1,n-1):
    ans=max(ans,dp1[i-1]+dp2[i+1])
print(ans)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
v=list(map(int,input().split(',')))
n=len(v)
dp1=[-float('inf')]*n
dp2=dp1[:]
dp1[0]=v[0]
dp2[-1]=v[-1]
for i in range(1,n):
    dp1[i]=max(dp1[i-1]+v[i],v[i])
    dp2[n-i-1]=max(dp2[n-i]+v[n-i-1],v[n-i-1])
ans=max(dp1)
for i in range(1,n-1):
    ans=max(ans,dp1[i-1]+dp2[i+1])
print(ans)
```

基本信息

#: 47586516

题目: 20744

提交人: 24n2400011782

内存: 9468kB

时间: 89ms

语言: Python3

提交时间: 2024-12-06 16:30:05

T25561: 2022决战双十一

brute force, dfs, <http://cs101.openjudge.cn/practice/25561/>

思路:

暴力,用递归回溯枚举购买方案,然后硬算.思路很简单,写起来挺恶心...

注意跨店满减是"每"满300-50

另外为什么示例里面同一家店会同时有满200-100和满250-80的券(

代码:

```
#处理输入
n,m=map(int,input().split())
goods=[]
for i in range(n):
    goods.append({})
    line=list(input().split())
    for _ in line:
        k,v=map(int,_.split(':'))
        goods[i][k]=v
coupons={}
for i in range(1,m+1):
    coupons[i]={}
    line=input().split()
    for _ in line:
        k,v=map(int,_.split('-'))
        coupons[i][k]=v
#构建对于每个商品的店铺列表(stores)和存储所有购物方案的列表(shopping_lists)
stores=[list(goods[i].keys()) for i in range(n)]
shopping_list=[0 for i in range(n)]
shopping_lists=[]

def buy(goods_id):
    global shopping_list,shopping_lists
    if goods_id==n:
        shopping_lists.append(shopping_list[:])
    else:
        for i in range(len(stores[goods_id])):
```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```
n,m=map(int,input().split())
goods=[]
for i in range(n):
    goods.append({})
    line=list(input().split())
    for _ in line:
        k,v=map(int,_.split(':'))
        goods[i][k]=v
coupons={}
for i in range(1,m+1):
    coupons[i]={}
    line=input().split()
    for _ in line:
        k,v=map(int,_.split('-'))
        coupons[i][k]=v
stores=[list(goods[i].keys()) for i in range(n)]
shopping_list=[0 for i in range(n)]
shopping_lists=[]

def buy(goods_id):
    global shopping_list,shopping_lists
    if goods_id==n:
        shopping_lists.append(shopping_list[:])
    else:
        for i in range(len(stores[goods_id])):
            shopping_list[goods_id]=i
            buy(goods_id+1)
        shopping_list[goods_id]=0
```

基本信息

#: 47620420

题目: 25561

提交人: 24n2400011782

内存: 4184kB

时间: 59ms

语言: Python3

提交时间: 2024-12-08 11:01:35

T20741: 两座孤岛最短距离

dfs, bfs, <http://cs101.openjudge.cn/practice/20741/>

思路:

bfs怎么优化都超时

看了某同学的解

还真不用bfs...

先dfs找岛,然后每两座岛之间计算所有陆地的(水平距离加竖直距离-1)取最小值

代码:

```
def dfs(x,y):#找岛,返回陆地的所有坐标
    global ans
    land=[(x,y)]
    d=[(-1,0),(1,0),(0,1),(0,-1)]
    for dx,dy in d:
        nx,ny=x+dx,y+dy
        if 0<=nx<n and 0<=ny<n and mat[nx][ny]=='1':
            mat[x][y]='0'
            land+=dfs(nx,ny)
        mat[x][y]='3'
    return set(land)

n=int(input())
mat=[]
islands=[]
for _ in range(n):
    mat.append(list(input()))
for i in range(n):
    for j in range(n):
        if mat[i][j]=='1':
```

```

        islands.append(dfs(i,j))
ans=float('inf')
for i in range(len(islands)):
    for j in range(i+1,len(islands)):
        island=list(islands[i])
        another=list(islands[j])
        ans=min(abs(island[a][0]-another[b][0])+abs(island[a][1]-another[b][1])-1 for a in range(len(island)) for b in range(len(another)))

```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```

def dfs(x,y):#找岛,返回陆地的所有坐标
    global ans
    land=[(x,y)]
    d=[(-1,0),(1,0),(0,1),(0,-1)]
    for dx,dy in d:
        nx,ny=x+dx,y+dy
        if 0<=nx<n and 0<=ny<n and mat[nx][ny]=='1':
            mat[x][y]='0'
            land+=dfs(nx,ny)
        mat[x][y]='3'
    return set(land)

n=int(input())
mat=[]
islands=[]
for _ in range(n):
    mat.append(list(input()))
for i in range(n):
    for j in range(n):
        if mat[i][j]=='1':
            islands.append(dfs(i,j))
ans=float('inf')
for i in range(len(islands)):
    for j in range(i+1,len(islands)):
        island=list(islands[i])
        another=list(islands[j])
        ans=min(abs(island[a][0]-another[b][0])+abs(island[a][1]-another[b][1])-1 for a in range(len(island)) for b in range(len(another)))
print(ans)

```

基本信息

#: 47589507
 题目: 20741
 提交人: 24n2400011782
 内存: 4324kB
 时间: 312ms
 语言: Python3
 提交时间: 2024-12-06 17:30:52

T28776: 国王游戏

greedy, <http://cs101.openjudge.cn/practice/28776>

思路:
 能猜...只用进行一些简单的数学推导,从只有两个大臣的情况开始想
 按(左手*右手)排序,别问我为什么(((
 早知道最后十几分钟写这个了
 代码:

```

n=int(input())
king=tuple(map(int,input().split()))
officials=[]
for i in range(n):
    officials.append(tuple(map(int,input().split())))
officials.sort(key=lambda x:x[0]*x[1])
pi=[king[0]]
for i in range(n):
    pi.append(officials[i][0]*pi[-1])
ans=king[0]//officials[0][1]
for i in range(1,n):
    ans=max(ans,pi[i]//officials[i][1])
print(ans)

```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
n=int(input())
king=tuple(map(int,input().split()))
officials=[]
for i in range(n):
    officials.append(tuple(map(int,input().split())))
officials.sort(key=lambda x:x[0]*x[1])
pi=[king[0]]
for i in range(n):
    pi.append(officials[i][0]*pi[-1])
ans=king[0]//officials[0][1]
for i in range(1,n):
    ans=max(ans,pi[i]//officials[i][1])
print(ans)
```

基本信息

#: 47590532
题目: 28776
提交人: 24n2400011782
内存: 3640kB
时间: 21ms
语言: Python3
提交时间: 2024-12-06 17:52:28

2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“计概2024fall每日选做”、CF、LeetCode、洛谷等网站题目。
奇怪的是我两道greedy都觉得不是很难...岛屿那题考试没做出来啊啊啊搜索和递归还是不熟,准备整理一下这部分的代码
今天试了用md格式做cheetsheet,上手之后真的比word好用和清楚得多...正在奋力将14页cheetsheet弄成md